Розробник:

Огонькова Наталія Максимівна

Група КІТ 101.8б

Варіант №11

**Лабораторна робота № 6**

Серіалізація/десеріалізація об’єктів. Бібліотека класів користувача

**Мета**:

* Тривале зберігання та відновлення стану об’єктів.
* Ознайомлення з принципами серіалізації/десеріалізації об’єктів.
* Використання бібліотек класів користувача.

**Вимоги:**

1. Реалізувати і продемонструвати тривале зберігання/відновлення раніше розробленого контейнера за допомогою серіалізації/десеріалізації.
2. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі л.р. №3 з іншим студентом (визначає викладач).
3. Продемонструвати послідовну та вибіркову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
4. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
5. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.
6. ОПИС ПРОГРАМИ
   1. Опис змінних

**boolean** loop = **true**; // змінна для масиву

**int** choose; // змінна для вибору користувача

**int** choose2; // змінна для вибору користувача

String str1; // змінна для вводу тексту

String str2; // змінна для вводу тексту

**int** temp = 0; // змінна для контейнера

1.2 Ієрархія та структура класів

**class** Lab06 – точка входу в програму

**class** Func – клас, що містить функції

**class** Container – клас, в якому містяться контейнери

**class** StringLenghtSort – клас, в якому міститься функція StringLenghtSort

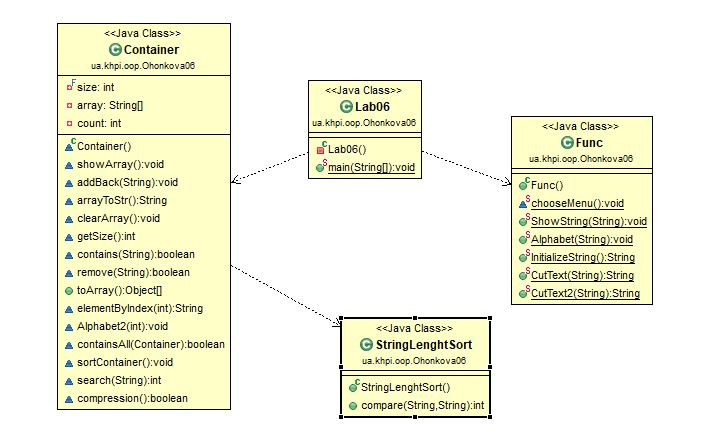


Рисунок 1 – Класи програми

* 1. Текст програми

**class** Lab06

**package** ua.khpi.oop.Ohonkova06;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.ObjectInputStream;

**import** java.io.ObjectOutputStream;

**import** java.util.Scanner;

**import** ua.khpi.oop.Ohonkova06.Container;

**final** **class** Lab06 {

**private** Lab06() {

//default constructor

}

/\*\*

\* **@param** args - ggg.

\* **@throws** IOException -ggg.

\* **@throws** ClassNotFoundException -ggg.

\*/

**public** **static** **void** main(**final** String[] args) **throws** IOException, ClassNotFoundException {

Container container = **new** Container();

Scanner scan = **new** Scanner(System.***in***);

**boolean** loop = **true**;

**int** choose;

**int** choose2;

String str1;

String str2;

**int** temp = 0;

**while** (loop) {

Func.*chooseMenu*();

choose = scan.nextInt();

**switch** (choose) {

**case** 1:

str1 = Func.*InitializeString*();

container.addBack(str1);

**break**;

**case** 2:

System.***out***.println("Your container:");

container.showArray();

**break**;

**case** 3:

System.***out***.println("\nCalculate:");

temp = container.getSize();

System.***out***.println("Enter number from 1 to " + temp);

choose2 = scan.nextInt();

container.Alphabet2(choose2);

System.***out***.println("Successfully!");

**break**;

**case** 4:

container.clearArray();

System.***out***.println("Clear container successfully!");

**break**;

**case** 5:

System.***out***.println("Container to String: ");

System.***out***.println(container.arrayToStr());

**break**;

**case** 6:

System.***out***.println("Sorting elements in container");

container.sortContainer();

System.***out***.println("Successfully");

**break**;

**case** 7:

str2 = Func.*InitializeString*();

**int** temper = container.search(str2);

**if** (temper == 0) {

System.***out***.println("This element is not abs");

**break**;

} **else** {

System.***out***.print("This elements contains in index: " + (temper - 1));

**break**;

}

**case** 8:

**if**(container.compression()) {

System.***out***.println("Elements are equal");

**break**;

} **else** {

System.***out***.println("Elements are not equal");

**break**;

}

**case** 9:

System.***out***.println("Saving to file");

FileOutputStream file = **new** FileOutputStream("tempFile.txt");

ObjectOutputStream object = **new** ObjectOutputStream(file);

object.writeObject(container);

object.close();

System.***out***.println("Successfully");

**break**;

**case** 10:

System.***out***.println("Read from file");

FileInputStream fileInput = **new** FileInputStream("tempFile.txt");

ObjectInputStream objectInput = **new** ObjectInputStream(fileInput);

container = (Container) objectInput.readObject();

objectInput.close();

System.***out***.println("Successfully");

**break**;

**case** 0:

System.***out***.println("\n\nThanks for working!");

loop = **false**;

**break**;

**default**:

System.***out***.println("Error. Mistake number!");

**break**;

}

}

}

}

**class** Func

**package** ua.khpi.oop.Ohonkova06;

**import** java.util.Scanner;

/\*

\* class Function in which all used functions are stored

\*

\* @author NATALIA

\* @version 5.0

\* Here start point of the program

\* \*/

**public** **class** Func {

/\*

\* Function that show user point of menu that he can choose

\* \*/

**static** **void** chooseMenu() {

System.***out***.println("\n\nChoose the right variant:");

System.***out***.println("1. Create and add string in container");

System.***out***.println("2. Show elements in container");

System.***out***.println("3. Count vowels and consonants");

System.***out***.println("4. Clear container");

System.***out***.println("5. Show container to string");

System.***out***.println("6. Sort");

System.***out***.println("7. Search");

System.***out***.println("8. Compression");

System.***out***.println("9. Save elements in file");

System.***out***.println("10. Read elements from file");

System.***out***.println("0. The End");

System.***out***.print("Write:");

}

/\*

\* Function show our string that we wrote

\* @param str is our string

\* \*/

**public** **static** **void** ShowString(String str){

System.***out***.println("Your string:");

System.***out***.println(str);

}

/\*

\* function alphabet counting vowel and other

\* @param v our vowels

\* \*/

**public** **static** **void** Alphabet (String str) {

**int** v = 0;

**int** sign = 0;

**char**[] vowels = {'a','e','i','o','u'};

**char** [] sentence = str.toLowerCase().toCharArray();

**for** (**char** letter : sentence) {

**if**(letter == ' ' || letter == '.' || letter == ',')

sign++;

**for** (**char** vowel : vowels) {

**if** (letter == vowel) {

v++;

}

}

}

System.***out***.println(str);

System.***out***.print("\tVowels:"+ v);

System.***out***.print("\tConsonants:" + (sentence.length -v-sign));

System.***out***.println();

System.***out***.println();

}

/\*

\* Function reads input string

\* @return str1 as our used string

\* \*/

**public** **static** String InitializeString(){

Scanner scan = **new** Scanner(System.***in***);

System.***out***.print("Enter text whith dots: ");

String str1 = **new** String();

str1 = scan.nextLine();

**return** str1;

}

/\*

\* Function truncates a string to a point and show the first sentence

\* @param index1 looking for dot

\* @return str

\* \*/

**public** **static** String CutText (String str) {

**int** index1=str.indexOf('.');

str = str.substring(index1+1);

**return** str;

}

/\*

\* Function truncates a string to a point and show the second sentence

\* @param index1 looking for dot

\* @return str

\* \*/

**public** **static** String CutText2 (String str) {

**int** index1=str.indexOf('.');

str = str.substring(0, index1+1);

**return** str;

}

}

**class** Container

**package** ua.khpi.oop.Ohonkova06;

**import** java.io.Serializable;

**import** java.util.Arrays;

**import** java.util.Iterator;

**import** java.util.NoSuchElementException;

**import** java.util.Scanner;

/\*\*

\* Contain elements in program.

\*/

**class** Container **implements** Serializable {

/\*\*

\* First size for array.

\*/

**private** **final** **int** size = 10;

/\*\*

\* Array contains all data.

\*/

**private** String[] array = **new** String[size];

/\*\*

\* Counter of number elements.

\*/

**private** **int** count = 0;

/\*\*

\* Showing array`s data.

\*/

**void** showArray() {

**if** (count == 0) {

System.***out***.println("Empty mass");

} **else** {

System.***out***.println();

**for** (**int** i = 0; i < count; i++) {

System.***out***.println(array[i]);

}

}

}

/\*\*

\* First size for array.

\* **@param** str1

\*/

**void** addBack(**final** String str1) {

**if** (count == array.length) {

array = Arrays.*copyOf*(array, array.length \* 2);

array[count++] = str1;

} **else** {

array[count++] = str1;

}

}

/\*\*

\* function translates an array to a string

\* **@param** str1 our future string

\*/

String arrayToStr() {

StringBuilder str1 = **new** StringBuilder("");

**if** (count != 0) {

str1 = **new** StringBuilder(array[0]);

str1.append(" ");

**for** (**int** i = 1; i < count; i++) {

str1.append(array[i]);

str1.append(" ");

}

}

**return** str1.toString();

}

**void** clearArray() {

array = **null**;

count = 0;

}

**int** getSize() {

**return** count;

}

/\*\*

\* the function checks if the element that is being passed is contained in the container

\*/

**boolean** contains(**final** String str) {

**boolean** cont = **false**;

**for** (**int** i = 0; i < count; i++) {

**if** (cont) {

**return** cont;

} **else** {

String str1;

str1 = array[i];

cont = str.equals(str1);

**if** (i == count - 1) {

**return** cont;

}

}

}

**return** cont;

}

/\*\*

\* removal of any element from the container

\*/

**boolean** remove(**final** String str) {

**boolean** remov = **false**;

**int** coun = 0;

**for** (**int** i = 0; i < count; i++) {

**if** (remov) {

**break**;

} **else** {

remov = str.equals(array[i]);

coun++;

**if** (i == count - 1) {

**break**;

}

}

}

array[coun - 1] = **null**;

**for** (**int** i = 0; i < count; i++) {

array[coun - 1] = array[coun++];

}

count--;

**return** remov;

}

**public** Object[] toArray() {

**if** (array == **null**) {

**return** **null**;

}

**return** Arrays.*copyOf*(array, count);

}

String elementByIndex(**final** **int** index) {

**return** array[index];

}

/\*

\* function alphabet counting vowel and other

\* @param v our vowels

\*\*/

**void** Alphabet2(**final** **int** index) {

String str2 = **new** String();

String str1 = **new** String();

str1 = Func.*CutText*(array[index - 1]);

str2 = Func.*CutText2*(array[index - 1]);

System.***out***.println("==============================");

Func.*Alphabet*(str2);

System.***out***.println("==============================");

Func.*Alphabet*(str1);

}

**boolean** containsAll(**final** Container container) {

**boolean** result;

**if** (container.getSize() != count) {

**return** **false**;

}

**for** (**int** i = 0; i < count; i++) {

result = array[i].equals(container.elementByIndex(i));

**if** (!result) {

**return** **false**;

}

}

**return** **true**;

}

**void** sortContainer() {

String[] newString;

newString = Arrays.*copyOf*(array, count);

StringLenghtSort str1 = **new** StringLenghtSort();

Arrays.*sort*(newString,str1);

array = Arrays.*copyOf*(newString, count);

}

**int** search(**final** String str) {

**boolean** cont = **false**;

**int** temp = 0;

**for** (**int** i = 0; i < count; i++) {

temp++;

**if** (cont) {

**return** temp;

} **else** {

String str1;

str1 = array[i];

cont = str.equals(str1);

**if** (i == count - 1) {

**if** (!cont) {

System.***out***.println("This element is abs");

**return** 0;

}

}

}

}

**return** temp;

}

**boolean** compression() {

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("Enter index first string: ");

**int** choose1 = scan.nextInt();

System.***out***.println("Enter index second string: ");

**int** choose2 = scan.nextInt();

**return** array[choose1-1].equals(array[choose2-1]);

}

}

**class** StringLenghtSort

**package** ua.khpi.oop.Ohonkova06;

**import** java.util.Comparator;

**public** **class** StringLenghtSort **implements** Comparator<String> {

/\*\*

\* **@param** o1 - first string.

\* **@param** o2 - second string.

\* **@return** - ggg.

\*/

@Override

**public** **int** compare(**final** String o1, **final** String o2) {

**return** Integer.*compare*(o1.length(), o2.length());

}

}

1. ВАРІАНТИ ВИКОРИСТАННЯ

Програму можна використовувати для підрахунку голосних і приголосних у тексті, яке задає користувач.

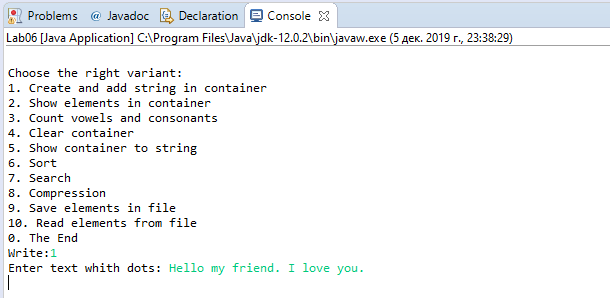


Рисунок 2 – Результат програми

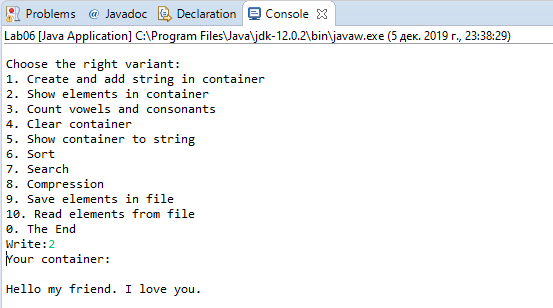


Рисунок 3 – Результат програми

ВИСНОВОК

При виконанні лабораторної роботи набуто практичних навичок, щодо тривалого зберігання та відновлення стану об’єктів, ознайомлення принципів серіалізації/десеріалізації об’єктів та використанні бібліотек класів користувача. Програма виконується без помилок.